

Table of Contents

1 MORE ON EXCEL UDF	9
1.1 PROCESS VECTOR / MATRIX.....	10
1.1.1 VECTOR / MATRIX AS INPUT	10
1.1.2 VECTOR / MATRIX AS OUTPUT	12
1.1.3 CONVERT TO A COLLECTION OF KEY-VALUE PAIRS	13
1.2 PROCESS DATE.....	14
1.2.1 OVERVIEW	14
1.2.2 A PRACTICAL EXAMPLE – <i>DATEADD()</i>	16
1.2.3 OTHER FEATURES (BUSINESS CALENDAR ETC)	21
1.3 HIDE “UNWANTED” FUNCTIONS IN EXCEL FUNCTION WIZARD.....	22
1.3.1 A SIMPLE SOLUTION	24
1.3.2 A BETTER SOLUTION	26
1.4 DEVELOP C# UDF WITHOUT VISUAL STUDIO	28
2 MORE ON SCRIPTING	31
2.1 DEBUGGING DYNAMIC SCRIPT.....	31
2.1.1 INTRODUCTION	31
2.1.1.1 SCRIPT DEBUGGING IN C++	31
2.1.1.2 SCRIPT DEBUGGING IN C#	34
2.1.1.3 MYUTIL.STARTDEBUG()	40
2.1.1.4 REVISED SCRIPT ENGINE IMPLEMENTATION	42
2.2 MULTIPLE SCRIPT SOURCE FILES.....	45
3 MORE ON C# / C++ INTEGRATION	47
3.1 DEBUG C#/C++ HYBRID APPLICATIONS.....	47

4	MORE ON THREADING	51
4.1	PROTECT CRITICAL RESOURCE	52
4.1.1	RACE CONDITION	52
4.1.2	USING <i>LOCK</i> STATEMENT	55
4.1.3	THE <i>INTERLOCKED</i> CLASS	56
4.1.4	THE <i>MONITOR</i> CLASS	57
4.1.5	THREAD RE-ENTRY	60
4.1.6	READER LOCK AND WRITER LOCK	61
4.2	AVOID DEADLOCK	67
4.3	LAZY CACHING AND DOUBLE-CHECKING	70
4.4	ELIMINATE UNNECESSARY LOCKING – A PRACTICAL EXAMPLE	75
4.5	DOUBLE-BUFFER PATTERN - A PRACTICAL EXAMPLE	78
4.6	THREAD SYNCHRONIZATION	83
4.6.1	SHARE DATA	85
4.6.2	SYNCHRONIZE EXECUTION FLOW	86
4.7	MONTE CARLO	95
4.8	INTER-PROCESS SYNCHRONIZATION	98
4.8.1	SYNCHRONIZE EXECUTION FLOW	98
4.8.2	SHARE DATA	101
4.9	DEBUG A MULTITHREADED APPLICATION	103
5	DEVELOP CLIENT APPLICATIONS	107
5.1	OVERVIEW AND DESIGN GUIDELINE	107
5.2	C# SOLUTION STRUCTURE.....	109
5.3	C# APPLICATION AS CLIENT APPLICATION	113
5.4	VB APPLICATION AS CLIENT APPLICATION.....	116
5.5	EXCEL AS CLIENT APPLICATION	120
5.6	WEB CLIENT	125
6	DEBUGGING TIPS	131
6.1	DISPLAY CUSTOM DEBUG INFORMATION	131

2.1.5. METHOD 1 – ToString()	133
2.1.6. METHOD 2 – DebuggerDisplay Attribute	134
2.1.7. METHOD 3 – AutoExp.cs	139
INDEX OF EXAMPLES	141
INDEX OF FIGURES	143